Function read PLC

%% lectura de bombillas

estBombilla1 = readBombilla1(m);

estBombilla2 = readBombilla2(m);

estBombilla3 = readBombilla3(m);

estBombilla4 = readBombilla4(m);

%% lectura electrovalvula

estQ17 = readQ17(m);

estEV = readEstEV(m);

%% lectura control del ventilador

estCViento = readCViento(m);

%% lectura sensores de cargas

readSCargaFuncs = {@readSCarga1, @readSCarga2, @readSCarga3, @readSCarga4};

for i = 1:4

estSCarga{i} = readSCargaFuncs{i}(m);

disp(estSCarga{i})

end

%% lectura Sensor de viento

estSViento = readSViento(m);

%% lectura Sensor de radiacion

estSRadiacion = readSRadiacion(m);

%% lectura sensores de humedad

estSHum1 = readHum1(m);

estSHum2 = readHum2(m);

estSTemp1 = readTemp1(m);

estSTemp2 = readTemp2(m);

%% lectura de los modos auto

estautoEV = readautoEV(m);

estreadAutoVent = readAutoVent(m);

estAutoLuces = readAutoLuces(m);

%% imprimir lecturas

disp("estBombilla1 =" + estBombilla1)

disp("estBombilla2 =" + estBombilla2)

disp("estBombilla3 =" + estBombilla3)

disp("estBombilla4 =" + estBombilla4)

disp("estCViento =" + estCViento +"V")

disp("estAutoEV =" + estautoEV)

disp("estQ17 =" + estQ17)

if estEV == 0

disp("La electrovalvula esta cerada")

end

if estEV == 1

disp("La electrovalvula esta abierta")

end

disp("estSViento =" + estSViento + "km/h")

disp("estSRadiacion =" + estSRadiacion)

disp("estAutoLuces =" + estAutoLuces)

disp("estreadAutoVent =" + estreadAutoVent)

disp("estSHum1 =" + estSHum1)

disp("estSHum2 =" + estSHum2)

disp("estSTemp1 =" + estSTemp1+"ºC")

disp("estSTemp2 =" + estSTemp2+"ºC")

Function Test

clear all;

clc

%% crear objeto modbus

try

m = modbus('tcpip','10.30.236.8',502,'Timeout',5,'NumRetries',2);

error = 0;

catch except

try

delete(m)

m = modbus('tcpip','10.30.236.8',502,'Timeout',5,'NumRetries',2);

error = 0;

catch exception

disp("Error al conectar el plc")

error = 1;

end

end

%% variables de control

u\_Bomb1 = 0;

u\_Bomb2 = 0;

u\_Bomb3 = 0;

u\_Bomb4 = 0;

u\_Viento = 0; % Voltage controlador (V) = Uviento/1000

u\_AutoEV = 0;

u\_AutoLuces = 1;

u\_AutoViento = 0;

%% proceso de control

if u\_Viento>4950

u\_Viento= 4950;

end

%% lectura de registros del PLC

disp("Estado inicial:")

read\_PLC

%% escritura de registros del PLC

writeBombilla1(m,u\_Bomb1)

writeBombilla2(m,u\_Bomb2)

writeBombilla3(m,u\_Bomb3)

writeBombilla4(m,u\_Bomb4)

writeCViento(m,u\_Viento)

writeAutoEV(m,u\_AutoEV)

writeAutoLuces(m,u\_AutoLuces)

writeAutoVent(m,u\_AutoViento)

%% Pulsos para apertura y cerada de la electrovalvula

writeQ17(m,1)

pause(0.05)

writeQ17(m,0)

read\_PLC

pause(5) %% temps en seconde

writeRiegoOFF(m,1)

pause(0.05)

writeRiegoOFF(m,0)

%volver a leer los datos

disp("Estado final:")

read\_PLC

delete(m)

Function pour lire entée digitale

function estBombilla1 = readBombilla1(m)

estBombilla1 = read(m,'coil',113,1);

end

Fonction pour lire une entrée analogique

function estSViento = readSViento(m)

estSViento\_bruto = read(m,'holdingregs',205,1,'uint16');

estSViento = estSViento\_bruto/100;

end

Fonction pour ecrire une valeur sur une sortie digitale

function writeBombilla1(m,u)

%m es el objet modbus y u la acción de control, 1 para encender, 0 para

%apagar

write(m,'coil',123,u);

end

Fonction pour ecrire sur une sortie analogique

function writeCViento(m,u)

%m es el objet modbus y u la acción de control, 1 para encender, 0 para

%apagar

write(m,'holdingregs',207,u) % sur Matalb on rajoute toujours 1 par rapport à l’adresse de la variable

end

Code de l’IHM app designer